CLAIMS

1. A method for the multi-exponentiation ($\Pi_{i=1}^{d} g_i^{e_i}$) or the multi-scalar multiplication ($\Sigma_{i=1}^{d} e_i \cdot g_i$) of elements ($g_i$) by means of in each case at least one exponent or scalar (e_i), in particular an integer exponent or scalar, which has in each case a maximum bit rate (n) or bit length, in particular for the exponentiation (g^e) or scalar multiplication ($e \cdot g$) of an element (g) by means of at least one exponent or scalar (e), in particular an integer exponent or scalar, which has in each case a maximum bit rate (n) or bit length, which elements (g_i; g) derive from at least one group (G), for example an Abelian group, which

- in the case of (multi-)exponentiation is notated in particular multiplicatively and
- in the case of (multi-)scalar multiplication is notated in particular additively,

characterized by the following method steps:

[a.1] computing and storing or

[a.2] retrieving from at least one memory

all powers (g_i^c) or all multiples ($c \cdot g_i$), wherein c is a permissible positive coefficient;

[b] dividing each exponent or scalar (e_i) into a number of chunks or into a number of parts ($e_{i,k}$) having a chunk or part width defined by a specific bit rate (L); and

[c] individually recoding the chunks or parts ($e_{i,k}$).

2. A method as claimed in claim 1, characterized in that the exponent or scalar (e_i) is represented in the divided form $e_i = \Sigma_{k=0}^{r} e_{i,k} 2^{kL}$, wherein

- r is defined as the number of chunks or parts ($e_{i,k}$), in particular as an integer quotient of the maximum bit rate (n) and the bit rate (L) of the chunk or part width, and
- $0 \leq e_{i,k} < 2^L$.

3. A method as claimed in claim 1 or 2, characterized in that the chunk or part

width (L) is selected to be

- significantly greater than a parameter (w) which corresponds to the width, in particular to the upper limit of the width, of a window over which the bits of the respective exponent or scalar (e_i) are read, and
- significantly shorter than the maximum length of each exponent or scalar (e_i),

in particular is selected prior to method step [a.1] and/or [a.2].

4. A method as claimed in at least one of claims 1 to 3, characterized in that

- in the case of (multi-)exponentiation, method step [c] of recoding the chunks or parts ($e_{i,k}$) can be divided into the following substeps for each individual chunk or for each individual part ($e_{i,k}$) of each exponent (e_i):

[c.1] setting a temporary variable (x) to a standardized value, in particular to the value 1 of the element of the group (G) which is neutral with respect to the group operation assigned to the group (G);

[c.2] successively setting a variable (k) to the values r-1, r-2, ..., 0, wherein for each value $k = r$-1, r-2, ..., 0 of the variable (k) the following substeps are carried out:

[c.2.i] for each value $i = 1, 2, ..., d$ of an index (i), wherein d is defined as the number of elements (g_i), in particular depending on the number of exponents (e_i) assigned to the elements (g_i):

[c.2.i.a] recoding the chunk or part ($e_{i,k}$) as the sum ($\Sigma_{j=0}{}^{L} b_{i,j}2^j$) of powers of two ($2^j$) weighted by in each case at least one coefficient ($b_{i,j}$) deriving from at least one finite set (C) of integers;

[c.2.i.b] if the coefficient ($b_{i,L}$) assigned to the highest power of two (2^L) does not vanish: setting the temporary variable (x) to the product of temporary variable (x) and the power ($g_i{}^{b_{i,L}}$) of the element (g_i) which is assigned to the coefficient ($b_{i,L}$) of the highest power of two (2^L);

[c.2.ii] for each value $j = L$-1, L-2, ..., 0 of the index (j):

[c.2.ii.a] squaring the temporary variable (x);

[c.2.ii.b] for each value $i = 1, 2, ..., d$ of the index (i):

if the coefficient ($b_{i,j}$) assigned to the power of two (2^j) does not vanish: setting the temporary variable (x) to the product of temporary variable (x) and the power ($g_i{}^{b_{i,j}}$) of the element (g_i) which is assigned to the respective coefficient ($b_{i,j}$) of the power of two (2^j); and

- after method step [c] of individually recoding the chunks or parts ($e_{i,k}$) the temporary variable (x) is returned.

5. A method as claimed in at least one of claims 1 to 3, characterized in that

- in the case of (multi-)scalar multiplication, method step [c] of recoding the chunks or parts ($e_{i,k}$) can be divided into the following substeps for each individual chunk or for each individual part ($e_{i,k}$) of each exponent (e_i):

[c.1] setting a temporary variable (x) to a standardized value, in particular to the value 0 of the element of the group (G) which is neutral with respect to the group operation assigned to the group (G);

[c.2] successively setting a variable (k) to the values r-1, r-2, ..., 0, wherein for each value $k = r$-1, r-2, ..., 0 of the variable (k) the following substeps are carried out:

[c.2.i] for each value $i = 1, 2, ..., d$ of an index (i), wherein d is defined as the number of elements (g_i), in particular depending on the number of scalars (e_i) assigned to the elements (g_i):

[c.2.i.a] recoding the chunk or part ($e_{i,k}$) as the sum ($\Sigma_{j=0}{}^{L} b_{i,j}2^j$) of powers of two ($2^j$) weighted by in each case at least one coefficient ($b_{i,j}$) deriving from at least one finite set (C) of integers;

[c.2.i.b] if the coefficient ($b_{i,L}$) assigned to the highest power of two (2^L) does not vanish: setting the temporary variable (x) to the sum of temporary variable (x) and the multiple ($b_{i,L}\cdot g_i$) of the element (g_i) which is assigned to the coefficient ($b_{i,L}$) of the highest power of two (2^L);

[c.2.ii] for each value $j = L$-1, L-2, ..., 0 of the index (j):

[c.2.ii.a] doubling the temporary variable (x);

[c.2.ii.b] for each value $i = 1, 2, ..., d$ of the index (i):

if the coefficient ($b_{i,j}$) assigned to the power of two (2^j) does not vanish: setting the temporary variable (x) to the sum of temporary variable (x) and the multiple ($b_{i,L}\cdot g_i$) of the element (g_i) which is assigned to the coefficient ($b_{i,j}$) of the power of two (2^j); and

- after method step [c] of individually recoding the chunks or parts ($e_{i,k}$) the temporary variable (x) is returned.

6. A method as claimed in at least one of claims 1 to 5, characterized in that

- the recoded chunk or the recoded part ($e_{i,k}$) is used once and
- the memory unit in which the recoded chunk or the recoded part ($e_{i,k}$) is stored is used to recode the following chunk or the following part ($e_{i,k-1}$).

7. A method as claimed in at least one of claims 1 to 6, characterized in that the method is implemented on at least one microprocessor assigned in particular to at least one chip card and/or in particular to at least one smart card.

8. A microprocessor which operates in accordance with a method as claimed in at least one of claims 1 to 7.

9. A device, in particular a chip card and/or in particular a smart card, having at least one microprocessor as claimed in claim 8.

10. The use of a method as claimed in at least one of claims 1 to 7 and/or of at least one microprocessor as claimed in claim 8 and/or of at least one device, in particular of at least one chip card and/or in particular of at least one smart card, as claimed in claim 9, in at least one cryptosystem, in particular in at least one public key cryptosystem, in at least one key exchange system or in at least one signature system.